

IMAGE SEGMENTATION ON CPU/GPU

Tomáš Bravenec

Bachelor Degree Programme (3), FEEC BUT

E-mail: xbrave01@stud.feec.vutbr.cz

Supervised by: Tomáš Frýza

E-mail: fryza@feec.vutbr.cz

Abstract: This article is focused on using graphical processing units for parallel data processing, specifically on image processing. Main focus of this thesis is determining time difference in image processing using graphical processing unit and classic approach on processor. Another focus is accessing webcam and processing of captured frames.

Keywords: image segmentation, parallel processing, GPU, NVIDIA, CUDA, image capture

1 ÚVOD

Článek se zabývá vytvořením aplikace pro segmentaci obrazu v programovacím jazyce C++ s využitím masivního paralelismu dostupného pomocí moderních grafických karet společnosti NVIDIA. Jelikož nároky na výpočetní výkon při zpracování obrazu narůstají exponenciálně s rostoucím rozlišením obrazu, roste stejně tak i doba nutná pro zpracování. Pro získání většího výpočetního výkonu lze využít výkonu grafických procesorů. K tomuto účelu existují dvě technologie, NVIDIA CUDA dostupná pouze na grafických kartách stejného výrobce a OpenCL dostupná na většině moderních grafických procesorů. Cílem aplikace je určit časový rozdíl v rychlosti zpracování pomocí běžného procesoru a dostupné grafické karty.

2 POPIS VYTVOŘENÉ APLIKACE

V rámci práce byly vytvořeny dvě aplikace pro segmentaci obrazu pomocí grafických karet. Jedna z těchto aplikací byla vytvořena pouze pro operační systémy Microsoft Windows Vista a novější, druhá aplikace je multiplatformní, má ovšem omezenou funkčnost proti aplikaci pro Windows.

Obě aplikace jsou vytvořeny ve vývojovém prostředí Microsoft Visual Studio. Aplikace pro Windows byla vytvořena pomocí a disponuje grafickým rozhraním (**Chyba! Nenalezen zdroj odkazů.**) a je schopná zpracovávat obraz pomocí procesoru i grafické karty, může načítat obraz ve formátu JPG, PNG a BMP nebo z připojené kamery. Naopak multiplatformní verze je konzolová aplikace a může zpracovávat obraz pouze pomocí grafických karet a obraz lze načíst jen z uloženého souboru ve stejných formátech jako ve verzi pro Windows. Obě aplikace využívají pro zpracování pomocí grafické karty technologii CUDA od společnosti NVIDIA a nelze je tudíž použít na grafických procesorech AMD a Intel.

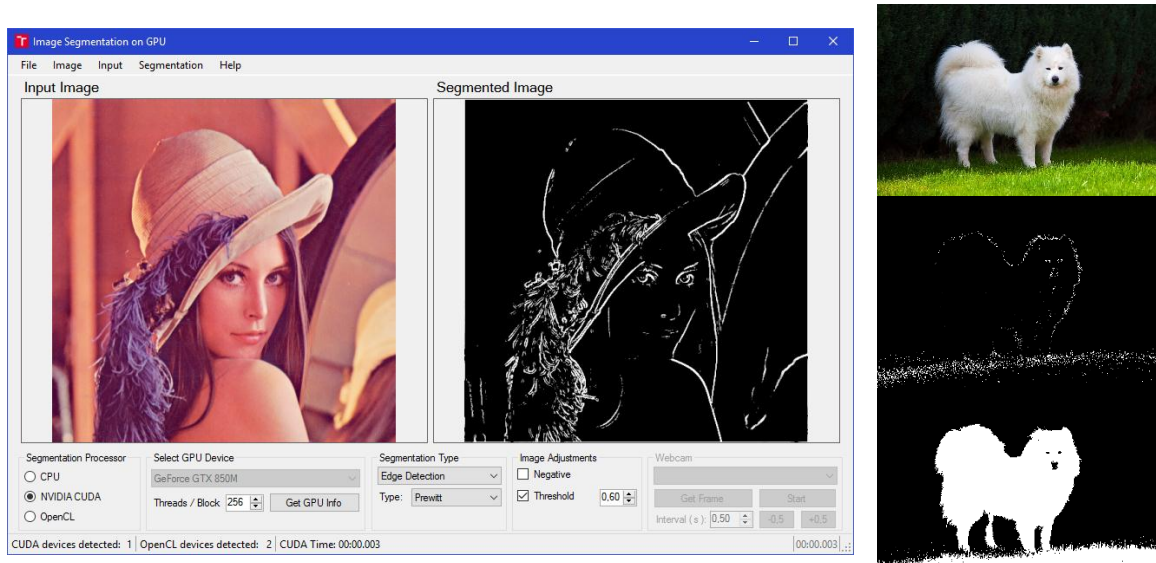
V současné verzi mají aplikace implementovány tyto metody zpracování obrazu:

- Metoda prahování ve třech provedeních
- Hranová detekce s třemi různými filtry

Spolu s těmito metodami aplikace provádějí měření doby zpracování a jsou schopné provést následnou úpravu hranové detekce prahováním pro odstranění většiny falešných hran a dosažení lepšího výsledku. U výsledného obrazu lze následně provést inverzi barev.

V budoucích verzích aplikace budou implementovány další segmentační metody jejichž algoritmus lze přepsat pro běh na grafických kartách, jednou z těchto metod je například shlukování k-means.

Pro objektivní srovnání výsledného času zpracování je pro zpracování pomocí běžného procesoru implementována technologie OpenMP pro paralelizaci zpracování pomocí více jader, což znamená využití plného potenciálu procesoru.



Obrázek 1: a) Uživatelské rozhraní aplikace pro operační systém Windows, b) vstupní obraz a zpracované výstupní obrazy hranovou detekcí a prahováním

2.1 METODA PRAHOVÁNÍ

Metoda prahování patří k nejsnazším metodám segmentace obrazu jak principem metody, tak z hlediska implementace algoritmu. Výstupem metody je binární obraz určující popředí a pozadí podle jasové složky obrazu. Při využití více prahového prahování je výstupem obraz v odstínech šedi. Metoda je vhodná převážně pro obrazy s vysokým kontrastem.

Podstatou metody je změnit jasovou hodnotu v obrazu, pokud je hodnota jasu pixelu nižší než daný práh, pixel je nahrazen černou barvou, a naopak pokud je vyšší než práh, pixel bude nahrazen barvou bílou. Při využití více prahového prahování se pro pixely spadající mezi dva prahy zbarví do šedé barvy. Hlavním problémem metody je výběr vhodného prahu, v případě výběru nevhodného prahu totiž může dojít k nerozeznatelnosti objektů na výstupním obrazu. Nejvhodnější obrazy pro metodu prahování jsou obrazy s vysokým kontrastem. [1]

2.2 HRANOVÁ DETEKCE

Hranová detekce má za úkol nalézt ve zpracovávaném obraze velké rozdíly v jasu, které značí hranu objektu nebo barvy. Existuje mnoho přístupů k detekci hran, nejsnadnějším je ale přístup pomocí derivací, kdy v případě první derivace se jedná o lokální extrém, nebo druhé derivace, kdy se jedná o průchod nulou. Tyto derivace lze aproximovat pomocí několika lokálních operátorů jako operátoru Sobelova, Prewittové a Kirschova. Po filtraci obrazu těmito operátory je následně nutné obraz prahovat pro odstranění falešných hran a šumu. [1]

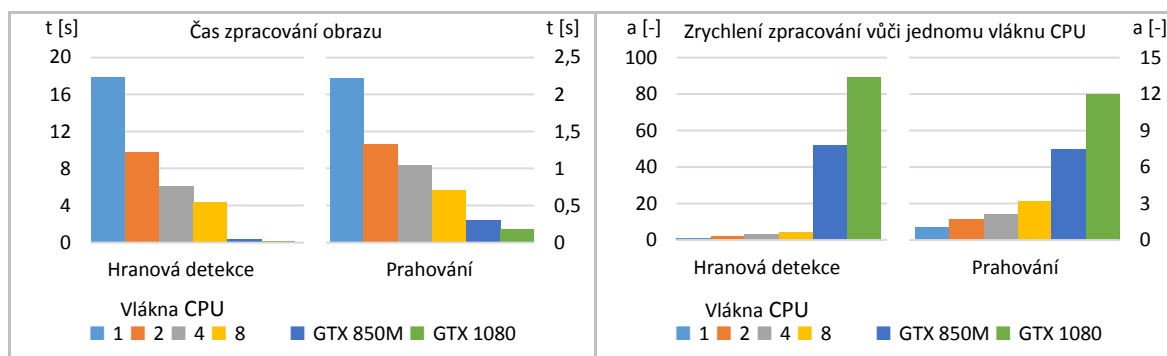
3 VYHODNOCENÍ

Aplikace byla testována na procesoru Intel Core i7 4700HQ se čtyřmi jádry (osmi vlákny) taktovanými na 3,4 GHz a grafických kartách NVIDIA GeForce GTX 850M obsahující 640 jader CUDA taktovaných na 0,9 GHz a NVIDIA GeForce GTX 1080 s 2560 CUDA jádry taktovanými na 1,86 GHz. Parametry obrazu a časy zpracování jsou v tabulce (Tabulka 1).

Metoda	Šířka	Výška	Počet pixelů	Intel Core i7 4700HQ				NVIDIA GeForce GTX 850M	NVIDIA GeForce GTX 1080
				Počet vláken CPU					
				1	2	4	8		
	P	P	MP	t [s]	t [s]	t [s]	t [s]	t [s]	t [s]
Hranová detekce	9402	6268	58,9	17,859	9,822	6,122	4,37	0,343	0,201
Prahování	9402	6268	58,9	2,219	1,336	1,042	0,707	0,299	0,186

Tabulka 1: Časy zpracování testovaného obrazu pro CPU a GPU

Naměřená data z tabulky jsou vyneseny do grafu (Obrázek 2a), jelikož jsou časy grafických karet v grafu pod hranicí rozeznatelnosti, na druhém grafu (Obrázek 2b) je zobrazeno relativní zrychlení zpracování vůči času jednoho jádra procesoru, v tomto případě lze těžko rozeznat zrychlení využití více vláken procesoru.



Obrázek 2: a) Graf časů zpracování, b) Graf zrychlení zpracování vůči jednomu vláknům CPU

Z grafu času zpracování hranové detekce lze vyčíst, že využití více vláken procesoru způsobí znatelné zrychlení zpracování, toto zrychlení ale ani zdaleka nedosahuje takových výsledků jako použití grafických karet, jejichž časy jsou z grafu nerozeznatelné. Naopak z grafu zrychlení je vidět jejich drtivá převaha nad procesorem. V případě metody prahování není rozdíl v rychlosti procesoru a grafických karet tak vysoký, to je ale způsobeno nižší náročností metody na výpočetní výkon oproti hranové detekci, i tak je ale urychlení zpracování citelné.

4 ZÁVĚR

Výsledky každé z implementovaných metod ve vytvořené aplikaci silně závisí na vlastnostech obrazu. Pro hranovou detekci i prahování existují vhodné i nevhodné obrazy, čímž je ovlivněn výsledek metod. Vhodně zvolený obraz s dobře zvolenými parametry pro danou metodu, má za důsledek kvalitní výstupní obraz obsahující buď nalezené hrany nebo objekty.

Z pohledu rychlosti zpracování je výsledek jednoznačný. Procesor s vyšším taktem, ale zato nižším počtem jader byl v každém případě pomalejší než grafická karta. Z toho vyplývá jasný výsledek, je-li implementovaná metoda správně napsána pro využívání grafických karet jako koprocessorů dojde téměř v každém případě k razantnímu urychlení běhu aplikace a úplně nezáleží na výkonové kategorii grafické karty, i mobilní grafické karty poskytnou výrazně lepší výsledky než procesory.

REFERENCE

- [1] Walek, P., Lamoš, M., Jan, J.: Analýza biomedicínských obrazů: Počítačová cvičení. Vysoké učení technické v Brně, 2013. ISBN 978-80-214-4792-9